AMENDMENT TO THE CLAIMS

1. (Currently Amended) A method of forming RLL coded data streams comprising:

dividing an input codeword into data portions and a separator portion;

placing the data portions into an output codeword with spaces between ~~each~~ the data ~~portion~~portions;

producing a separator matrix from the separator portion, the separator matrix being composed of a plurality rows, each row being a separator sub-matrix of ones and zeros such that each row is nonzero; and

stuffing a respective one of the plurality of rows into a respective one of the ~~space~~ spaces between the~~each~~ data ~~portion~~ portions of the input codeword to form ~~an~~ the output codeword.

2. (Original) The method of claim 1 wherein separator sub-matrices prevent the output codeword from having more than a predetermined number of consecutive zeros.

3. (Original) The method of claim 1 wherein the step of producing comprises:

passing the separator portion to an encoder to produce an encoded separator portion; and

generating separator blocks of a predetermined bit size from the encoded separator portion.

4. (Original) The method of claim 1 wherein the data portions of the input codeword are placed directly into the output codeword without encoding.

5. (Currently Amended) The method of claim 1 wherein the data portions of the input codeword are groups of bits, each group having a predetermined number of bits.

6. (Currently Amended) The method of claim 1 further comprising:

interleaving the separator ~~blocks~~ sub-matrices without changing boundaries between the separator sub-matrices ~~blocks~~ prior to stuffing.

7. (Original) The method of claim 1 further comprising:

changing an order of data portions of equal size within the output codeword without changing boundaries between data portions.

8. (Currently amended) A method of forming RLL coded data streams, the method comprising:

separating an input data block into data blocks, each data block having one or more data bits;

dividing one of the data blocks into a plurality of sets of data, each set having a predetermined number of bits;

encoding the sets of data in an encoder to form separator blocks; and

forming an output code word from the data blocks and the separator blocks such that the separator blocks are positioned between the data blocks within the codeword.

9. (Currently Amended) The method of claim 8, wherein ~~the~~ each RLL coded data stream has a component code rate of 10/11 and a k-constraint of no more than 12 consecutive zeros.

10. (Original) The method of claim 8 further comprising:

permuting the separator blocks after encoding.

11. (Original) The method of claim 8 further comprising:

permuting the data blocks and the separator blocks separately before forming the output code word.

12. (Original) The method of claim 8 wherein a binary value of each separator block is greater than zero.

13. (Currently Amended) The method of claim 8 wherein ~~the~~ a code rate of the RLL ~~code~~coded data stream is 48/49.

14. (Currently Amended) A system for producing a coded data stream having consecutive one values separated by a separator block, the system comprising:

an RLL encoder adapted to separate an input code word into data portions and a separator portion, the RLL encoder adapted to place the data portions into an output codeword with ~~space~~ spaces between ~~each~~ the data ~~portion~~ portions;

an encoder block adapted to process the separator portion into a separator matrix and adapted to place a respective one of the rows of the separator matrix into a respective one of the ~~space~~ spaces between ~~each~~ the data ~~portion~~ portions in the output codeword; and

a transceiver adapted to transmit the output codeword to a channel.

15. (Currently Amended) The system of claim 14, further comprising:

front end and timing elements for filtering data read from the ~~sub~~channel;

a decoder block for processing the output codeword into data portions and rows of the separator ~~portions~~ matrix and for decoding the separator ~~portions~~ matrix; and

a RLL decoder for decoding the data portions.

16. (Original) The system of claim 14 wherein the system is a disc drive.

17. (Original) The system of claim 14 and further comprising:

an interleaver adapted to process the output codeword prior to transmission by the transceiver.

18. (Currently Amended) The system of claim 17 wherein the output codeword is interleaved without changing boundaries between the data portions in the output codeword.

19. (Currently Amended) A method for encoding data for transmission over a channel, the method comprising:

breaking an input codeword into *n* data portions and a separator portion;

placing the data portions into an output codeword without encoding, each data portion being separated from a next data portion by a space;

encoding the separator portion into *n* minus 1 separator blocks; and

placing a respective one of the *n* minus 1 separator ~~block in~~ blocks into a respective one of the ~~space~~ spaces between the data portions in the output codeword.

20. (Original) The method of claim 19 further comprising:

writing the output codeword to the channel.

21. (Currently Amended) The method of claim 19 wherein one separator block of the *n* minus 1 separator blocks has a fewest number of bits compared with other separator blocks, the step of placing the separator blocks further comprising:

placing the one separator block having the fewest number of bits between data portion (n) and data portion (n - 1).

22. (Currently Amended) The method of claim 19 and further comprising:

permuting the n minus 1 separator blocks with a first encoder, and

permuting the n data blocks with a second encoder[[B]].

23. (Original) The method of claim 19 further comprising:

detecting transmitted data using an iterative detection scheme.